

## **Zadanie – Python (poziom średni)**

Temat: Analiza podróży samochodowej – dataclass, walidacja, funkcje, obsługa wyjątków

Celem zadania jest zaprojektowanie i zimplementowanie prostego modułu analitycznego w Pythonie,  
który oblicza zużycie paliwa oraz koszt przejazdu na podstawie danych wejściowych.

Zadanie sprawdza umiejętności:

- pracy z dataclass
- tworzenia i używania funkcji
- walidacji danych wejściowych
- obsługi wyjątków (try / except)

### **1. Model danych (dataclass)**

Zdefiniuj klasę danych (dataclass) reprezentującą pojedynczą podróż samochodową.

Klasa powinna zawierać co najmniej następujące pola:

- dystans\_km (float) – długość trasy w kilometrach
- srednie\_spalanie\_na\_100km (float) – średnie spalanie pojazdu (l / 100 km)
- cena\_paliwa\_za\_litr (float) – cena paliwa za 1 litr

Wymagania:

- typy pól muszą być jawnie określone
- wartości pól powinny być walidowane (np. brak wartości ujemnych)

### **2. Funkcje obliczeniowe**

Napisz niezależne funkcje realizujące logikę biznesową.

Wymagane funkcje:

- 1) Funkcja obliczająca zużycie paliwa na podstawie dystansu i średniego spalania
- 2) Funkcja obliczająca koszt przejazdu na podstawie zużycia paliwa i ceny paliwa

Wymagania:

- funkcje nie mogą korzystać ze zmiennych globalnych
- każda funkcja powinna posiadać docstring
- funkcje powinny zwracać wartości typu float

### **3. Walidacja danych**

Zaimplementuj walidację danych wejściowych.

Walidacja powinna obejmować:

- sprawdzenie typów danych (np. int / float)
- sprawdzenie zakresu wartości (np. brak wartości ujemnych, brak zera tam, gdzie nie ma sensu)
- zgłaszanie odpowiednich wyjątków (ValueError, TypeError)

Walidacja może być realizowana:

- w konstruktorze dataclass
- w funkcjach
- lub w obu miejscach

### **4. Obsługa wyjątków (try / except)**

Zaimplementuj fragment programu, który:

- tworzy obiekt podróży
- wywołuje funkcje obliczeniowe
- przechwytuje i obsługuje wyjątki za pomocą bloku try / except

W przypadku błędnych danych program:

- nie powinien się wywrać
- powinien wyświetlić czytelny komunikat o błędzie

### **5. Scenariusz testowy**

Przetestuj rozwiązanie dla następującego scenariusza:

- dystans: 350 km
- średnie spalanie: 7.2 l / 100 km
- cena paliwa: 6.85 zł / l

Program powinien:

- obliczyć zużycie paliwa
- obliczyć koszt przejazdu
- zaprezentować wynik w czytelnej formie

### **6. Wymagania końcowe**

- kod powinien być czytelny i zgodny z PEP8
- nazwy zmiennych i funkcji w języku polskim lub angielskim (konsekwentnie)

- brak kodu w sekcji globalnej poza uruchomieniem programu

**Zadanie rozszerzające (dla chętnych)**

- obsługa listy wielu podróży
- zsumowanie kosztów i zużycia paliwa
- zaokrąglanie wyników do 2 miejsc po przecinku
- przygotowanie słownikowej struktury wyników